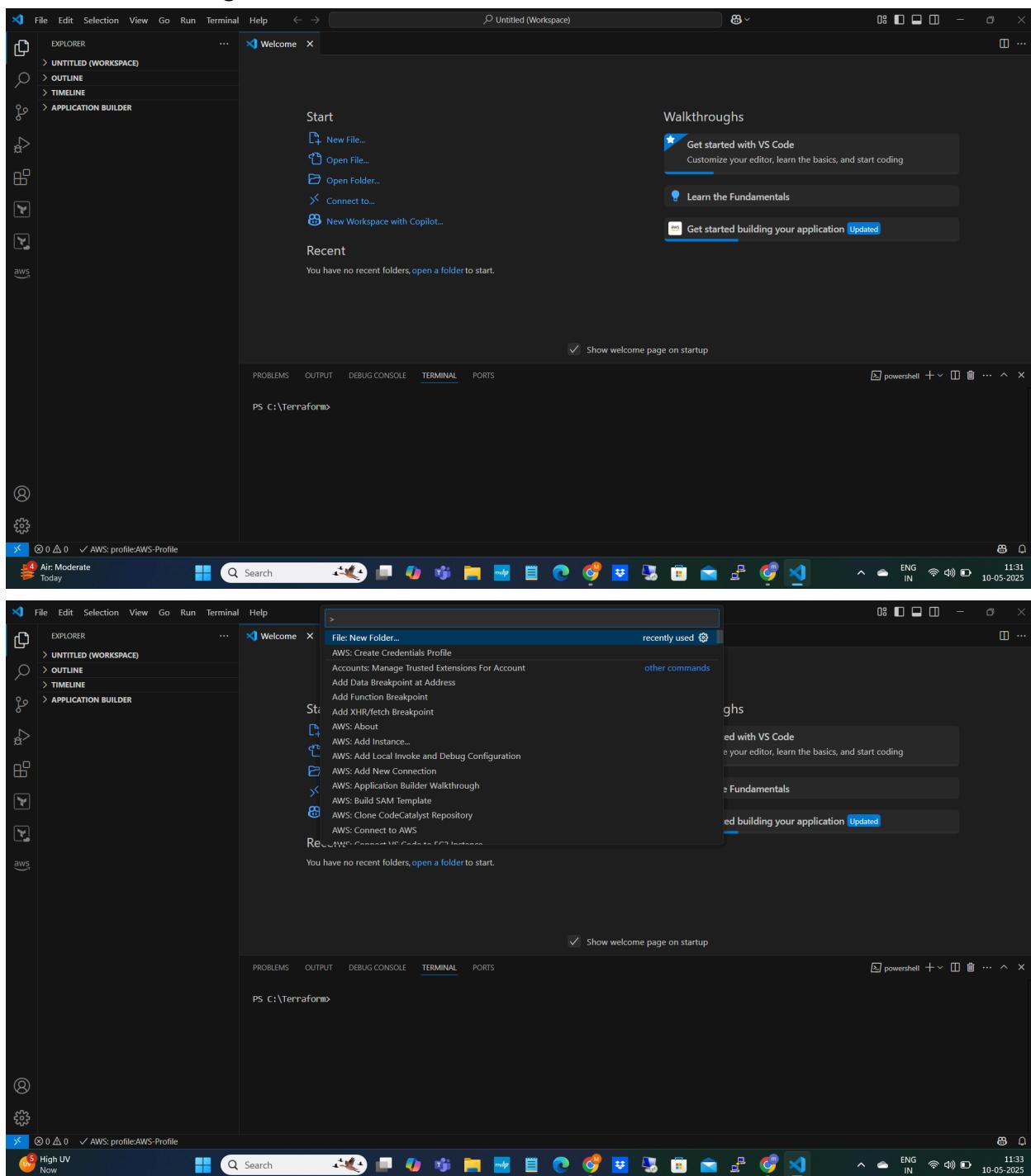
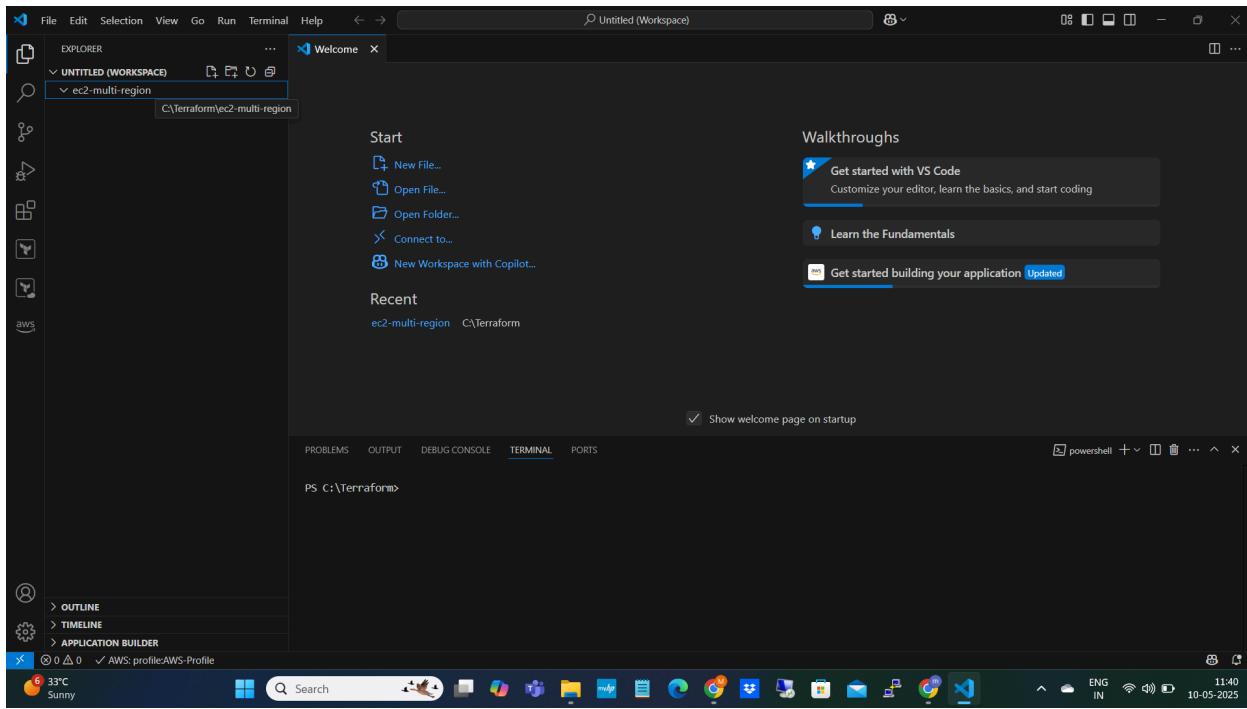


Terraform Task-2

Creating Folder in Visual studio to create terraform files

Folder: **ec2-multi-region**





To Create 2 EC2 Instances required 2 key pairs in 2 different regions and Access Key Accesskey:

The screenshot shows the AWS IAM console with the following details:

- Left Sidebar:** Identity and Access Management (IAM) dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), Access reports (Access Analyzer, External access, Unused access, Analyzer settings, Credential report, Organization activity).
- Middle Content:**
 - Multi-factor authentication (MFA) (1):** A table showing one virtual MFA device assigned to the user.
 - Access keys (2):** A table showing two access keys:

Access key ID	Created on	Access key last used	Region last used	Service last used	Status
AKIA5FTZBLOBFWX4POPW	46 days ago	9 days ago	us-east-1	sts	Inactive
AKIA5FTZBLOBNNFHIZI4	9 days ago	1 hour ago	us-east-1	sts	Active
 - CloudFront key pairs (0):** No CloudFront key pairs listed.
- Bottom:** CloudShell, Feedback, and system status bar showing the date and time (13:14 10-05-2025).

Creating 2 Key pairs for 2 regions in AWS Account:

1. Us-east-1: us-east-1-key-pair

The screenshot shows the AWS CloudShell interface. In the top navigation bar, the URL is `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateKeyPair`. The main content area is titled "Key pair" and describes it as a set of security credentials for connecting to an instance. A form is filled out with the following details:

- Name:** us-east-1-key-pair
- Key pair type:** RSA (selected)
- Private key file format:** .pem (selected)
- Tags - optional:** No tags associated.
- Add new tag:** A button to add up to 50 more tags.

At the bottom right of the form are "Cancel" and "Create key pair" buttons. The status bar at the bottom of the screen shows "CloudShell Feedback" and the date "10-05-2025".

Click on Create Key pair and it is created successfully

The screenshot shows the AWS CloudShell interface. In the top navigation bar, the URL is `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#KeyPairs`. The main content area displays a success message: "Successfully created key pair". Below this, a table lists the key pairs:

Name	Type	Created	Fingerprint	ID
us-east-1-key-pair	rsa	2025/05/10 11:58 GMT+5:30	27:66:e5:1a:10:d1:6f:0d:30:14:8c:35:81:d1:9c...	key-0b56d..
aws-key	rsa	2025/04/08 22:58 GMT+5:30	5fc1:36:c7:27:9f:2d:75:d0:be:8a:3e:6d:28:28:...	key-05532..

The left sidebar shows the EC2 navigation menu with sections like Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The status bar at the bottom shows "CloudShell Feedback" and the date "10-05-2025".

Changed the region and creating the new key pair

2. Us-west-2: us-west-2-key-pair

The screenshot shows the AWS EC2 console with the 'Key pairs' page open. The left sidebar shows navigation options like Dashboard, Instances, Images, and Elastic Block Store. The main area displays a table with columns: Name, Type, Created, and Fingerprint. A search bar at the top says 'Find Key Pair by attribute or tag'. On the right, a large list of AWS Regions is shown, grouped by continent:

Region	Region ID
United States	us-east-1, us-east-2, us-west-1, us-west-2
Asia Pacific	ap-south-1, ap-northeast-3, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1
Canada	ca-central-1
Europe	eu-central-1, eu-west-1, eu-west-2, eu-west-3, eu-north-1

At the bottom right of the region list are 'Manage Regions' and 'Manage Local Zones' buttons.

The screenshot shows the 'Create key pair' wizard. Step 1: Key pair details. It includes fields for Name (set to 'us-west-2-key-pair'), Key pair type (set to RSA), Private key file format (.pem selected), and Tags - optional (with an 'Add new tag' button). The status bar at the bottom indicates the date and time as 10-05-2025 12:01.

Click on Create Key pair and it is created successfully

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar navigation includes 'Instances', 'Images', and 'Elastic Block Store'. The main content area displays a table titled 'Key pairs (1/1)'. A green banner at the top indicates 'Successfully created key pair'. The table has columns for 'Name', 'Type', 'Created', 'Fingerprint', and 'ID'. One row is listed: 'us-west-2-key-pair' (rsa), created on '2025/05/10 12:02 GMT+5:30', with a fingerprint '7df0:3e:49:ef:2a:8a:79:84:e6:fe:8b:4f:a8:bb...' and an ID 'key-0ccc2...'. Action buttons for 'Actions' and 'Create key pair' are visible.

Now Create the variables.tf File inside the folder ec2-multi-region

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The left sidebar shows an 'EXPLORER' view with a folder named 'ec2-multi-region' containing a file named 'variables.tf'. The main workspace shows a 'Start' section with options like 'New File...', 'Open File...', and 'Open Folder...'. A 'Walkthroughs' section includes links to 'Get started with VS Code', 'Learn the Fundamentals', and 'Get started building your application'. The bottom status bar shows the path 'C:\Terraform>' and the profile 'AWS: profile.AWS-Profile'. The taskbar at the bottom includes icons for various Windows applications like File Explorer, Task View, and the Start button.

Define the Variables:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a workspace named "UNTITLED (WORKSPACE)" containing a folder "ec2-multi-region" and a file "variables.tf".
- Editor:** Displays the content of "variables.tf":

```
variable "instance_type" {
  default = "t2.micro"
}

variable "east_key_name" {
  description = "Key Pair name for us-east-1"
  default     = "us-east-1-key-pair"
}

variable "west_key_name" {
  description = "Key Pair name for us-west-2"
  default     = "us-west-2-key-pair"
}
```

- Terminal:** Shows the command "PS C:\Terraform>" indicating the current working directory.
- Bottom Status Bar:** Includes icons for weather (34°C, Sunny), search, taskbar, and system status (language EN IN, battery 12:05, date 10-05-2025).

Create the main.tf file

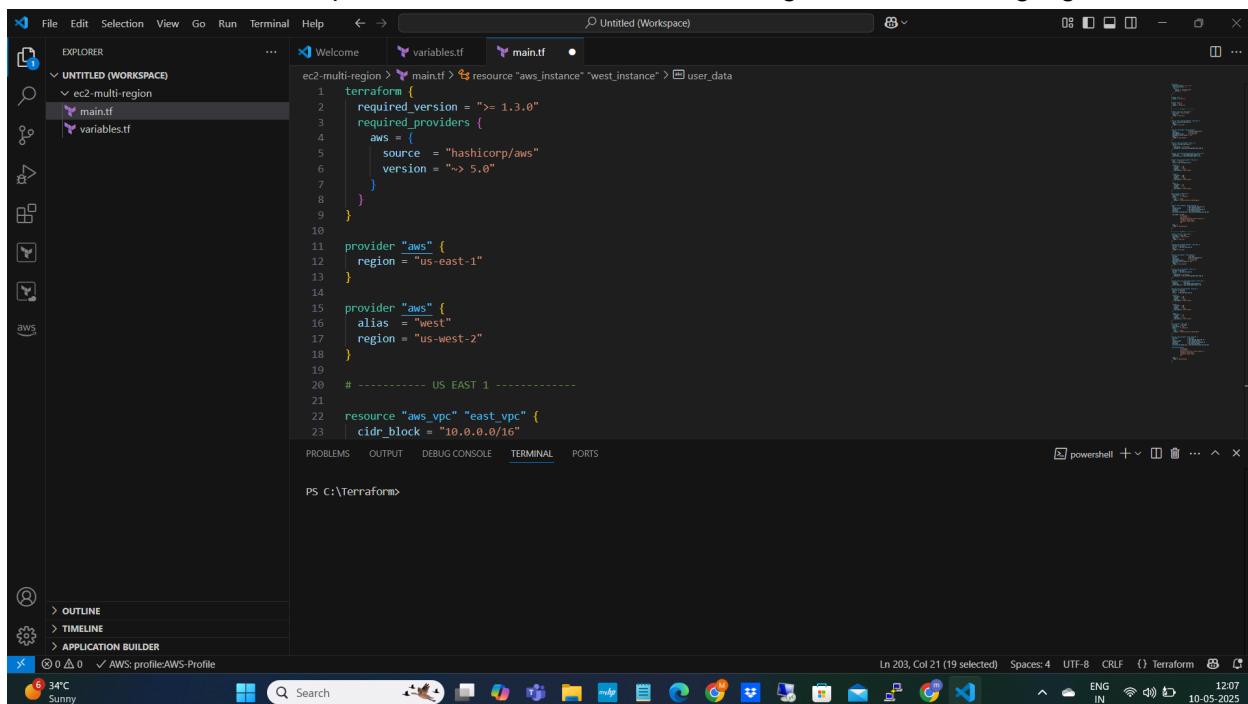
The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a workspace named "UNTITLED (WORKSPACE)" containing a folder "ec2-multi-region" and files "main.tf" and "variables.tf".
- Editor:** Displays the content of "main.tf":

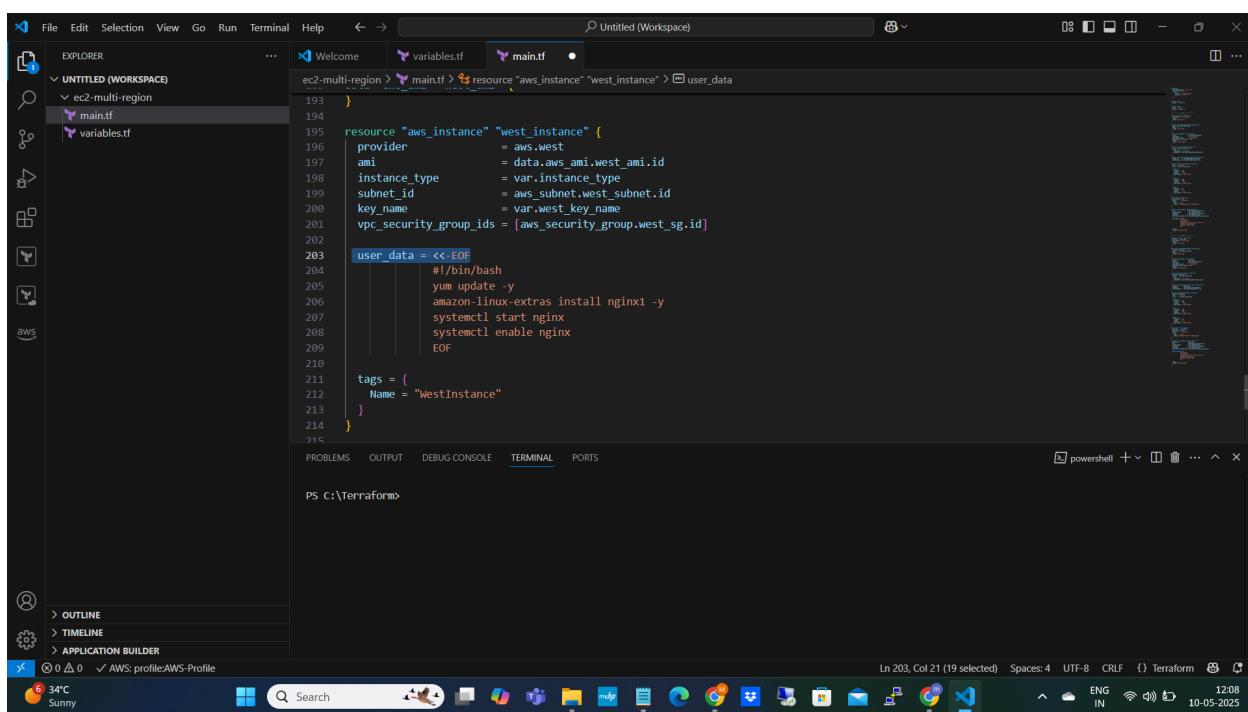
```
Open chat (Ctrl+U), or select a language (Ctrl+K M), or fill with template to get started.  
Start typing to dismiss or don't show this again.
```

- Terminal:** Shows the command "PS C:\Terraform>" indicating the current working directory.
- Bottom Status Bar:** Includes icons for weather (34°C, Sunny), search, taskbar, and system status (language EN IN, battery 12:06, date 10-05-2025).

Provide the terraform script to Create 2 EC2 instance in 2 regions and installing Nginx on them



```
1 terraform {
2     required_version = ">= 1.3.0"
3     required_providers {
4         aws = {
5             source  = "hashicorp/aws"
6             version = ">= 5.0"
7         }
8     }
9 }
10
11 provider "aws" {
12     region = "us-east-1"
13 }
14
15 provider "aws" {
16     alias  = "west"
17     region = "us-west-2"
18 }
19
20 # ----- US EAST 1 -----
21
22 resource "aws_vpc" "east_vpc" {
23     cidr_block = "10.0.0.0/16"
```



```
193 }
194
195 resource "aws_instance" "west_instance" {
196     provider      = aws.west
197     ami           = data.aws_ami.west_ami.id
198     instance_type = var.instance_type
199     subnet_id    = aws_subnet.west_subnet.id
200     key_name     = var.west_key_name
201     vpc_security_group_ids = [aws_security_group.west_sg.id]
202
203     user_data = <<EOF
204         #!/bin/bash
205         yum update -y
206         amazon-linux-extras install nginx1 -y
207         systemctl start nginx
208         systemctl enable nginx
209         EOF
210
211     tags = [
212         {Name = "WestInstance"}
213     ]
214 }
```

Create [output.tf](#) file:

The screenshot shows the Visual Studio Code interface with a dark theme. In the Explorer sidebar, there is a workspace named 'UNTITLED (WORKSPACE)' containing files: 'main.tf', 'outputs.tf', and 'variables.tf'. The 'outputs.tf' file is open in the main editor area, displaying the following Terraform code:

```
1 output "east_instance_public_ip" {
2   value = aws_instance.east_instance.public_ip
3 }
4
5 output "west_instance_public_ip" {
6   value = aws_instance.west_instance.public_ip
7 }
```

The terminal at the bottom shows the command 'PS C:\Terraform>'.

Terraform Already installed in my system

Let's check the version: `$ terraform version`

The screenshot shows the Visual Studio Code interface with a dark theme. In the Explorer sidebar, there is a workspace named 'UNTITLED (WORKSPACE)' containing files: 'main.tf', 'outputs.tf', and 'variables.tf'. The 'main.tf' file is open in the main editor area, displaying the following Terraform code:

```
193 }
194
195 resource "aws_instance" "west_instance" {
196   provider      = aws.west
197   ami           = data.aws_ami.west_ami.id
198   instance_type = var.instance_type
199   subnet_id     = aws_subnet.west_subnet.id
200   key_name      = var.west_key_name
201   vpc_security_group_ids = [aws_security_group.west_sg.id]
202
203   user_data = <<-EOF
204     #!/bin/bash
205     yum update -y
206     amazon-linux-extras install nginx1 -y
207     systemctl start nginx
208     systemctl enable nginx
209     EOF
210
211   tags = [
212     { Name = "WestInstance" }
213   ]
214 }
```

The terminal at the bottom shows the command being run: 'PS C:\Terraform> terraform version'. The output of the command is:

```
PS C:\Terraform> terraform version
Terraform v1.11.4
on windows_amd64
+ provider registry.terraform.io/hashicorp/aws v5.96.0
PS C:\Terraform>
```

The status bar at the bottom indicates the file is 'profile.AWS-Profile'.

Now Configure the Aws : \$ aws --version

A screenshot of the Visual Studio Code interface. The left sidebar shows an 'EXPLORER' view with a workspace named 'UNTITLED (WORKSPACE)' containing files 'main.tf', 'outputs.tf', and 'variables.tf'. The main code editor window displays Terraform code for creating an EC2 instance in the 'west' region. Below the code editor is a terminal window showing the output of the 'aws --version' command:

```
PS C:\Terraform> PS C:\Terraform> terraform version
Terraform v1.11.4
on windows_amd64
+ provider registry.terraform.io/hashicorp/aws v5.96.0
PS C:\Terraform> aws --version
aws-cli/2.25.1 Python/3.12.9 Windows/11 exe/AMD64
PS C:\Terraform>
```

The status bar at the bottom indicates the date and time as '10-05-2025'.

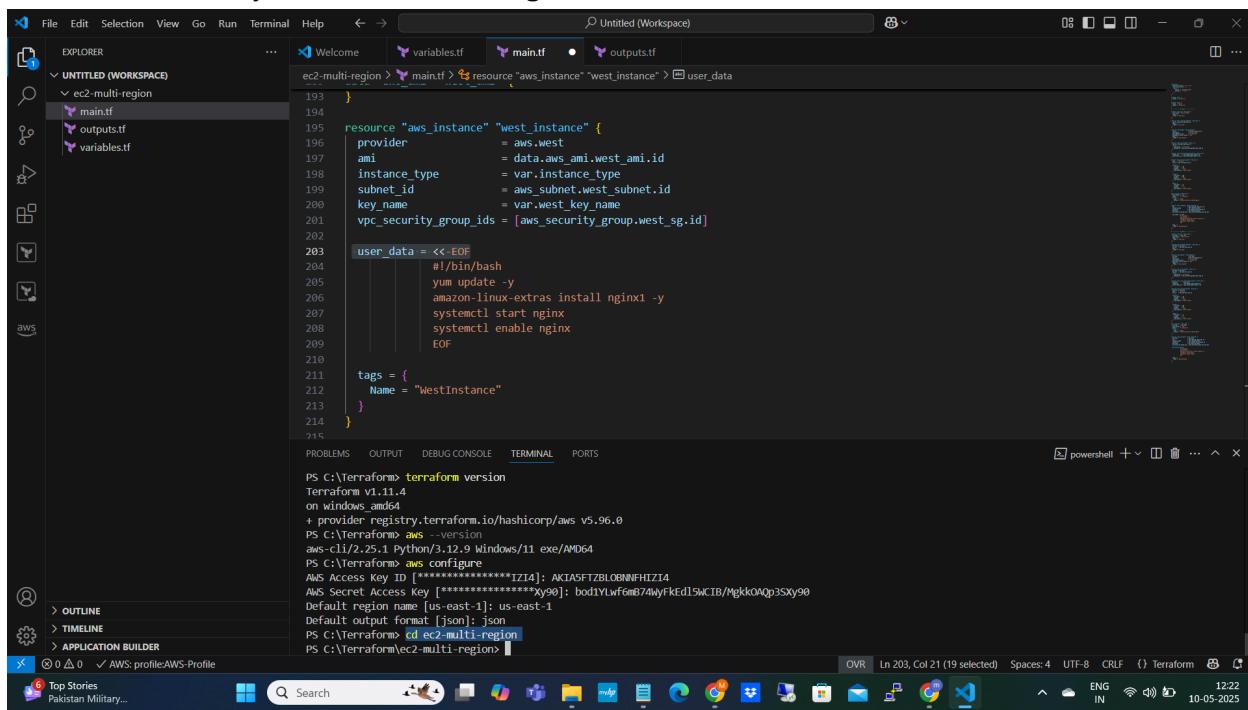
\$ aws configure :

A screenshot of the Visual Studio Code interface, identical to the previous one but with a different terminal command. The terminal window now shows the output of the 'aws configure' command:

```
PS C:\Terraform> PS C:\Terraform> terraform version
Terraform v1.11.4
on windows_amd64
+ provider registry.terraform.io/hashicorp/aws v5.96.0
PS C:\Terraform> aws --version
aws-cli/2.25.1 Python/3.12.9 Windows/11 exe/AMD64
PS C:\Terraform> aws configure
AWS Access Key ID [*****]: AKIA5FTZBLOBNNFHIZI4
AWS Secret Access Key [*****]: b0d1YLwfmB74wyfKEdl5WCIB/Mgkk0Qp3Sxy90
Default region name [us-east-1]: us-east-1
Default output format [json]: json
PS C:\Terraform>
```

The status bar at the bottom indicates the date and time as '12:20'.

Access the directory : \$ cd ec2-multi-region



```
resource "aws_instance" "west_instance" {
  provider      = aws.west
  ami           = data.aws_ami.west_ami.id
  instance_type = var.instance_type
  subnet_id     = aws_subnet.west_subnet.id
  key_name      = var.west_key_name
  vpc_security_group_ids = [aws_security_group.west_sg.id]

  user_data = <<-EOF
    #!/bin/bash
    yum update -y
    amazon-linux-extras install nginx1 -y
    systemctl start nginx
    systemctl enable nginx
  EOF

  tags = [
    { Name = "WestInstance" }
  ]
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Terraform> terraform version
Terraform v1.11.4
on windows_amd64
+ provider registry.terraform.io/hashicorp/aws v5.96.0
PS C:\Terraform> aws --version
aws-cli/2.25.1 Python/3.12.9 Windows/11 exe/AMD64
PS C:\Terraform> aws configure
AWS Access Key ID [*****]: AKIA5FTZBLOBNFHIZI4
AWS Secret Access Key [*****]: b0diYLwGmB74yfKEd15WCIB/MgkkO4Qp3Sxy90
Default region name [us-east-1]: us-east-1
Default output format [json]: json
PS C:\Terraform> cd ec2-multi-region
PS C:\Terraform\ec2-multi-region>
```

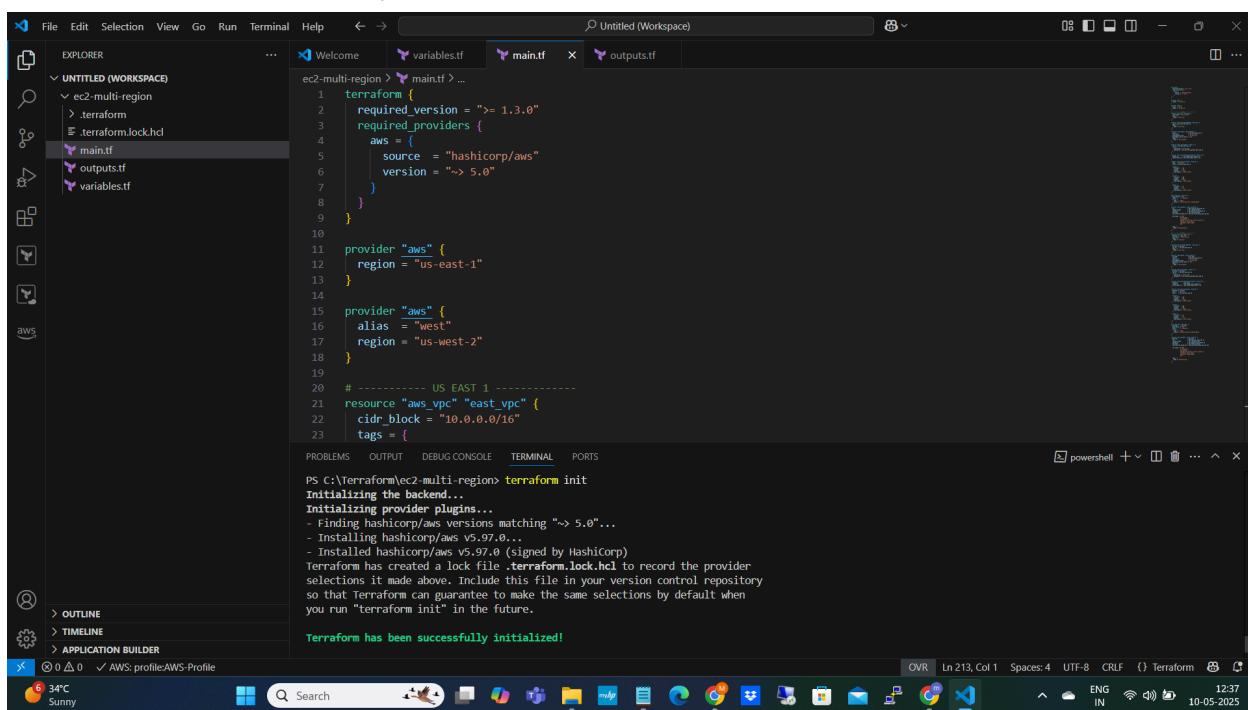
OVR Ln 203, Col 21 (19 selected) Spaces:4 UTF-8 CRLF {} Terraform

Top Stories Pakistan Military...

Search

12:22 10-05-2025

Now Initialize the Terraform: \$ terraform init



```
terraformer {
  required_version = ">= 1.3.0"
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "> 5.0"
    }
  }
}

provider "aws" {
  region = "us-east-1"
}

provider "aws" {
  alias  = "west"
  region = "us-west-2"
}

# ----- US EAST 1 -----
resource "aws_vpc" "east_vpc" {
  cidr_block = "10.0.0.0/16"
  tags = {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Terraform\ec2-multi-region> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "> 5.0"...
- Installing hashicorp/aws v5.97.0...
- Installed hashicorp/aws v5.97.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!
```

OVR Ln 213, Col 1 Spaces:4 UTF-8 CRLF {} Terraform

34°C Sunny

Search

12:37 10-05-2025

\$ terraform validate

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command "terraform validate" being run in a PowerShell session. The output shows that Terraform has been successfully initialized and that the configuration is valid.

```
PS C:\Terraform\ec2-multi-region> terraform validate
Success! The configuration is valid.
```

Lets display the plan before execution:

\$ terraform plan

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command "terraform plan" being run in a PowerShell session. The output shows the execution plan, indicating that resources will be created. Terraform used the selected providers to generate the execution plan, and it will perform actions such as creating a new AWS AMI.

```
PS C:\Terraform\ec2-multi-region> terraform plan
data.aws_ami.east_am: Reading...
data.aws_ami.west_am: Reading...
data.aws_ami.east_am: Read complete after 2s [id=ami-097947612b141c026]
data.aws_ami.west_am: Read complete after 2s [id=ami-08a0cbdcdbc088448]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```

The screenshot shows the AWS Cloud9 IDE interface with the terminal tab active. The terminal output shows the results of running `terraform validate` and `terraform plan` on a workspace named "ec2-multi-region".

```
PS C:\Terraform\ec2-multi-region> terraform validate
Success! The configuration is valid.

PS C:\Terraform\ec2-multi-region> terraform plan
data.aws_ami.east_ami: Reading...
data.aws_ami.west_ami: Reading...
data.aws_ami.east_ami: Read complete after 2s [id=ami-097947612b141c026]
data.aws_ami.west_ami: Read complete after 2s [id=ami-08a0cbcdcbc08448]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.east_instance will be created
+ resource "aws_instance" "east_instance" {
    ami = "ami-097947612b141c026"
    arn = (known after apply)
    associate_public_ip_address = (known after apply)
    availability_zone = (known after apply)
    cpu_core_count = (known after apply)
    cpu_threads_per_core = (known after apply)
    disable_api_stop = (known after apply)
    disable_api_termination = (known after apply)
    ebs_optimized = (known after apply)
    enable_primary_ipv6 = (known after apply)
    get_password_data = false
    host_id = (known after apply)
    host_resource_group_arn = (known after apply)
    iam_instance_profile = (known after apply)
    id = (known after apply)
    instance_initiated_shutdown_behavior = (known after apply)
    instance_lifecycle = (known after apply)
    instance_state = (known after apply)
    instance_type = "t2.micro"
    ipv4_address_count = (known after apply)
    ipv4_addresses = "us-east-1 key-pair"
    monitoring = (known after apply)
    outpost_arn = (known after apply)
    password_data = (known after apply)
}
```

This screenshot is identical to the one above, showing the AWS Cloud9 IDE terminal output for validating and planning a Terraform configuration across two AWS regions.

```
PS C:\Terraform\ec2-multi-region> terraform validate
Success! The configuration is valid.

PS C:\Terraform\ec2-multi-region> terraform plan
data.aws_ami.east_ami: Reading...
data.aws_ami.west_ami: Reading...
data.aws_ami.east_ami: Read complete after 2s [id=ami-097947612b141c026]
data.aws_ami.west_ami: Read complete after 2s [id=ami-08a0cbcdcbc08448]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.west_instance will be created
+ resource "aws_instance" "west_instance" {
    ami = "ami-08a0cbcdcbc08448"
    arn = (known after apply)
    associate_public_ip_address = (known after apply)
    availability_zone = (known after apply)
    cpu_core_count = (known after apply)
    cpu_threads_per_core = (known after apply)
    disable_api_stop = (known after apply)
    disable_api_termination = (known after apply)
    ebs_optimized = (known after apply)
    enable_primary_ipv6 = (known after apply)
    get_password_data = false
    host_id = (known after apply)
    host_resource_group_arn = (known after apply)
    iam_instance_profile = (known after apply)
    id = (known after apply)
    instance_initiated_shutdown_behavior = (known after apply)
    instance_lifecycle = (known after apply)
    instance_state = (known after apply)
    instance_type = "t2.micro"
    ipv4_address_count = (known after apply)
    ipv4_addresses = "us-west-2 key-pair"
    monitoring = (known after apply)
    outpost_arn = (known after apply)
    password_data = (known after apply)
    placement_group = (known after apply)
    placement_partition_number = (known after apply)
    primary_network_interface_id = (known after apply)
}
```

The screenshot shows the AWS Cloud9 IDE interface with the following details:

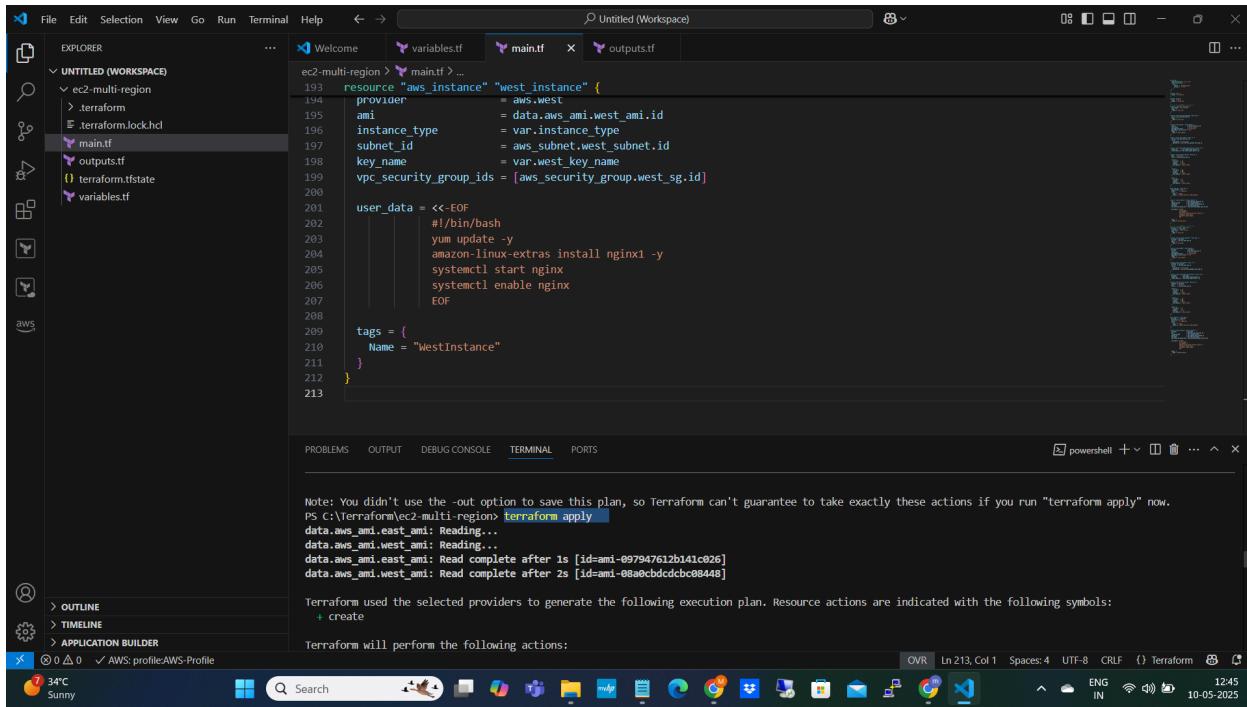
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS.
- Terminal Tab:** powershell + (with icons for copy, paste, search, etc.).
- Code Area:** Displays Terraform code for creating two AWS Subnets: `aws_subnet.east_subnet` and `aws_subnet.west_subnet`. The code includes resource definitions for `arn`, `assign_ipv6_address_on_creation`, `availability_zone`, `availability_zone_id`, `cidr_block`, `enable_dns64`, `enable_resource_name_dns_a_record_on_launch`, `enable_resource_name_dns_aaaa_record_on_launch`, `id`, `ipv6_cidr_block_association_id`, `ipv6_native`, `map_public_ip_on_launch`, `owner_id`, `private_dns_hostname_type_on_launch`, and `tags`.
- Bottom Status Bar:** Shows AWS profile:AWS-Profile, 34°C Sunny, and system status including battery level, network, and date/time (10-05-2025).

The screenshot shows the AWS Cloud9 IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS.
- Terminal Tab:** powershell + (with icons for copy, paste, search, etc.).
- Code Area:** Displays Terraform code for creating an AWS VPC named `west_vpc`. The code includes resource definitions for `main_route_table_id`, `owner_id`, and `tags`. It also includes a note about `aws_vpc.west_vpc` being created and defines its properties like `arn`, `cidr_block`, `default_network_acl_id`, `default_route_table_id`, `default_security_group_id`, `dhcp_options_id`, `enable_dns_hostnames`, `enable_dns_support`, `enable_network_address_usage_metrics`, `id`, `instance_tenancy`, `ipvs_association_id`, `ipvs_cidr_block`, `ipvs_cidr_block_network_border_group`, `main_route_table_id`, `owner_id`, and `tags`.
- Bottom Status Bar:** Shows AWS profile:AWS-Profile, 34°C Sunny, and system status including battery level, network, and date/time (10-05-2025).

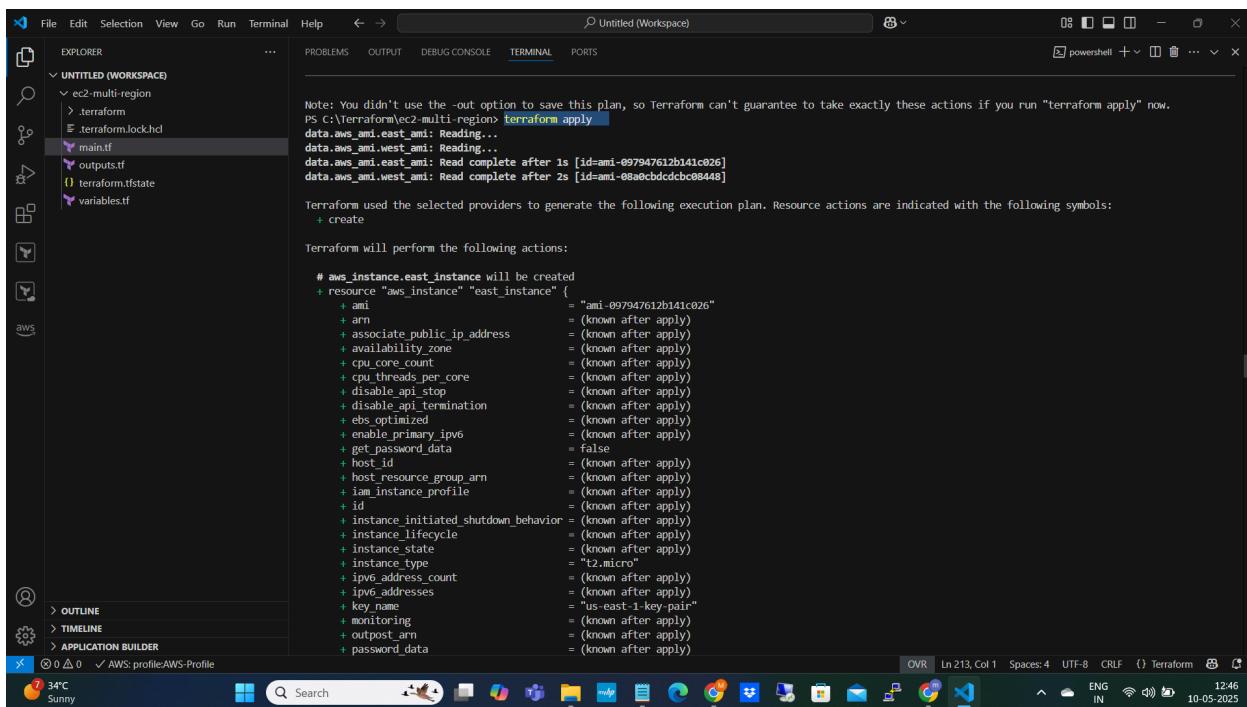
Now Execute the plan by executing apply command

\$ terraform apply



The screenshot shows the VS Code interface with the following details:

- File Structure:** The left sidebar shows an "EXPLORER" view with files: ".terraform", ".terraform.lock.hcl", "main.tf", "outputs.tf", and "variables.tf".
- Main Editor:** The main editor pane displays Terraform code for creating EC2 instances in two regions (East and West). The code includes provider declarations, resource definitions for AWS instances, and user data scripts.
- Terminal:** The bottom terminal shows the command "terraform apply" being run in a PowerShell window. It outputs resource reads and completion messages.
- Status Bar:** The status bar at the bottom indicates the environment is "AWS: profile/AWS-Profile".



The screenshot shows the VS Code interface with the following details:

- File Structure:** The left sidebar shows an "EXPLORER" view with files: ".terraform", ".terraform.lock.hcl", "main.tf", "outputs.tf", and "variables.tf".
- Main Editor:** The main editor pane displays Terraform code for creating EC2 instances in two regions (East and West). The code includes provider declarations, resource definitions for AWS instances, and user data scripts.
- Terminal:** The bottom terminal shows the command "terraform apply" being run in a PowerShell window. It outputs resource reads and completion messages.
- Status Bar:** The status bar at the bottom indicates the environment is "AWS: profile/AWS-Profile".

Entered Value as : yes to execute apply

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of a Terraform apply command. The user has entered 'yes' at the prompt 'Enter a value: yes'. The terminal output shows the creation of various AWS resources, including VPCs, subnets, security groups, and route tables, along with their corresponding resource IDs.

```
+ tags
+   "Name" = "west-vpc"
}
+ tags_all
+   "Name" = "west-vpc"
}

Plan: 14 to add, 0 to change, 0 to destroy.

Changes to outputs:
+ east_instance_public_ip = (known after apply)
+ west_instance_public_ip = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.east_vpc: Creating...
aws_vpc.west_vpc: Creating...
aws_vpc.east_vpc: Creation complete after 4s [id=vpc-037faf9c0cc18a864]
aws_internet_gateway.east_igw: Creating...
aws_subnet.east_subnet: Creating...
aws_security_group.east_sg: Creating...
aws_vpc.west_vpc: Creation complete after 4s [id=vpc-07f6ccabdb59b84e]
aws_internet_gateway.west_igw: Creating...
aws_subnet.west_subnet: Creating...
aws_security_group.west_sg: Creating...
aws_internet_gateway.east_igw: Creation complete after 1s [id=igw-0abffcb8ef674e2a3]
aws_route_table.east_rt: Creating...
aws_internet_gateway.west_igw: Creation complete after 2s [id=igw-0ec3d311268439f77]
aws_route_table.west_rt: Creating...
aws_route_table.east_rt: Creation complete after 3s [id=rtb-0bd0e9bcbcc4219aa]
aws_security_group.east_sg: Creation complete after 5s [id=sg-090a520a11993f79a]
aws_route_table.west_rt: Creation complete after 2s [id=rtb-0c49e8211c4fee569]
aws_security_group.west_sg: Creation complete after 5s [id=sg-09ddc2fb24b6d59c7]
aws_subnet.east_subnet: Still creating... [10s elapsed]
aws_subnet.west_subnet: Still creating... [10s elapsed]
aws_subnet.east_subnet: Creation complete after 12s [id=subnet-08d9ed5879de294a3]
```

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of a Terraform apply command, showing the successful creation of resources. The output includes resource IDs and a summary message indicating 14 resources added, 0 changed, and 0 destroyed. The status bar at the bottom right shows the date and time as 10-05-2025.

```
aws_vpc.east_vpc: Creating...
aws_vpc.west_vpc: Creating...
aws_vpc.east_vpc: Creation complete after 4s [id=vpc-037faf9c0cc18a864]
aws_internet_gateway.east_igw: Creating...
aws_subnet.east_subnet: Creating...
aws_security_group.east_sg: Creating...
aws_vpc.west_vpc: Creation complete after 4s [id=vpc-07f6ccabdb59b84e]
aws_internet_gateway.west_igw: Creating...
aws_subnet.west_subnet: Creating...
aws_security_group.west_sg: Creating...
aws_internet_gateway.east_igw: Creation complete after 1s [id=igw-0abffcb8ef674e2a3]
aws_route_table.east_rt: Creating...
aws_internet_gateway.west_igw: Creation complete after 2s [id=igw-0ec3d311268439f77]
aws_route_table.west_rt: Creating...
aws_route_table.east_rt: Creation complete after 3s [id=rtb-0bd0e9bcbcc4219aa]
aws_security_group.east_sg: Creation complete after 5s [id=sg-090a520a11993f79a]
aws_route_table.west_rt: Creation complete after 2s [id=rtb-0c49e8211c4fee569]
aws_security_group.west_sg: Creation complete after 5s [id=sg-09ddc2fb24b6d59c7]
aws_subnet.east_subnet: Still creating... [10s elapsed]
aws_subnet.west_subnet: Still creating... [10s elapsed]
aws_subnet.east_subnet: Creation complete after 12s [id=subnet-08d9ed5879de294a3]
aws_route_table_association.east_rta: Creating...
aws_instance.east_instance: Creating...
aws_route_table_association.east_rta: Creation complete after 1s [id=rtbassoc-0a6a7be61f81636c6]
aws_subnet.west_subnet: Creation complete after 12s [id=subnet-0797ef1807a6b1bf9]
aws_route_table_association.west_rta: Creating...
aws_instance.west_instance: Creating...
aws_route_table_association.west_rta: Creation complete after 1s [id=rtbassoc-0e6a46f621b6f6e938]
aws_instance.east_instance: Still creating... [10s elapsed]
aws_instance.west_instance: Still creating... [10s elapsed]
aws_instance.east_instance: Creation complete after 16s [id=i-022c5a42215de5747]
aws_instance.west_instance: Creation complete after 16s [id=i-0b8c295fdb99ce1d9]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.

Outputs:

east_instance_public_ip = "54.161.127.28"
west_instance_public_ip = "35.94.124.45"
PS C:\Terraform\ec2-multi-region>
```

Apply execution done successfully

Now let's check the AWS console whether the 2 Ec2 instances created in two regions:

The screenshot shows the AWS EC2 home page in the us-east-1 region. The left sidebar includes links for Dashboard, Instances (selected), Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays various AWS resources with their counts: Instances (running) 1, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 1, Key pairs 2, Load balancers 0, Placement groups 0, Security groups 3, Snapshots 0, and Volumes 1. There are sections for Launch instance, Service health (AWS Health Dashboard), and Explore AWS (Introducing Spot Blueprints). The status bar at the bottom indicates it's 12:48 on 10-05-2025.

EastInstance created in us-east-1 region:

The screenshot shows the AWS EC2 Instances page for the us-east-1 region. The left sidebar shows the EC2 navigation path. The main content displays a table of instances with one entry: EastInstance (i-022c5a42215de5747). The instance details are: Instance ID i-022c5a42215de5747, Instance state Running, Instance type t2.micro, Status check 2/2 checks passed, Alarm status None, Availability Zone us-east-1a, and Public IPv4 - (not assigned). The Details tab is selected, showing the Instance summary and Instance details. The status bar at the bottom indicates it's 12:49 on 10-05-2025.

The screenshot shows the AWS EC2 Instances details page for instance `i-022c5a42215de5747`. The instance is running and has a public IPv4 address of `54.161.127.28`. It is associated with a VPC ID `vpc-037faf9c0cc18a864` and a subnet ID `subnet-08d9ed5879de294a3`. The instance type is `t2.micro`. The instance ARN is `arn:aws:ec2:us-east-1:1905418201986:instance/i-022c5a42215de5747`.

us-east-1: Vpc,subnets and Internet gateway:

vpc:

The screenshot shows the AWS VPC dashboard. There are two VPCs listed: `east-vpc` (VPC ID `vpc-037faf9c0cc18a864`) and another VPC (VPC ID `vpc-07f2dc389716a7f01`). Both VPCs are in an available state with their Block Public Access setting set to off. The IPv4 CIDR ranges are `10.0.0.0/16` and `172.31.0.0/16` respectively.

VPC dashboard < Actions

vpc-037faf9c0cc18a864 / east-vpc

Details		Info	
VPC ID	vpc-037faf9c0cc18a864	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-05426e2e526587c2e	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-0284c3b1a4690244f7
		IPv6 pool	-
		Owner ID	905418201986

Resource map | CIDs | Flow logs | Tags | Integrations

Resource map Info

VPC	Subnets	Route tables	Network
Show details	Subnets within this VPC	Route tables (2)	Connect
Your AWS virtual network	us-east-1a	rtb-0bd089bcbcc4219aa	east-ic
east-vpc			

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 13:23 10-05-2025

Subnet:

VPC dashboard < Actions

Subnets (1/7) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-07c77b859be636e7c	Available	vpc-07f2dc389716a7f01	Off	172.31.48.
-	subnet-08d6a59da11aa09b	Available	vpc-07f2dc389716a7f01	Off	172.31.32.
-	subnet-086c940b64a97a6d4	Available	vpc-07f2dc389716a7f01	Off	172.31.00.
-	subnet-087f1289ec00d8051	Available	vpc-07f2dc389716a7f01	Off	172.31.64.
-	subnet-0f67565ca4322f8a2	Available	vpc-07f2dc389716a7f01	Off	172.31.16.
<input checked="" type="checkbox"/> east-subnet	subnet-08d9ed5879de294a3	Available	vpc-037faf9c0cc18a864 east-v...	Off	10.0.1.0/2
-	subnet-0e07ae4a5177b4d083	Available	vpc-07f2dc389716a7f01	Off	172.31.80.

subnet-08d9ed5879de294a3 / east-subnet

Details		Flow logs		Route table		Network ACL		CIDR reservations		Sharing		Tags	
Subnet ID	subnet-08d9ed5879de294a3	Subnet ARN	arn:aws:ec2:us-east-1:905418201986:subnet/subnet-08d9ed5879de294a3	State	Available	Block Public Access	Off						
IPv4 CIDR	10.0.1.0/24	IPv6 CIDR	-	IPv6 CIDR association ID	-								
Availability Zone	us-east-1a	Available IPv4 addresses	10.0.1.0 - 10.0.1.254	Network border group	vpc-037faf9c0cc18a864	VPC	vpc-07f2dc389716a7f01						

https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SubnetData... © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 13:24 10-05-2025

Screenshot of the AWS VPC Subnet Details page for subnet-08d9ed5879de294a3.

Details

Subnet ID	subnet-08d9ed5879de294a3	Subnet ARN	arn:aws:ec2:us-east-1:905418201986:subnet-08d9ed5879de294a3	State	Available	Block Public Access	Off
IPv4 CIDR	10.0.1.0/24	Available IPv4 addresses	250	IPv6 CIDR	-	IPv6 CIDR association ID	-
Availability Zone	us-east-1a	Availability Zone ID	use1-az6	Network border group	us-east-1	VPC	vpc-037faf9c0cc18a864 east-vpc
Route table	rtb-0bd089bcbcc4219aa	Auto-assign IPv6 address	No	Default subnet	No	Auto-assign public IPv4 address	Yes
IPv4 CIDR reservations	-	Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-	Outpost ID	-
Resource name DNS A record	Disabled	IPv6 CIDR reservations	-	IPv6-only	No	Hostname type	IP name
				DNS64	Disabled	Owner	905418201986
				Resource name DNS AAAA record	Disabled		

Actions

Flow logs Route table Network ACL CIDR reservations Sharing Tags

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 34°C Partly sunny 13:24 10-05-2025

internet gateway:

Screenshot of the AWS VPC Internet Gateways page.

Internet gateways (1/2) info

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-01907f92c32f10b85	Attached	vpc-07f2dc389716a7f01	905418201986
<input checked="" type="checkbox"/> east-igw	igw-0abffcb8ef674e2a3	Attached	vpc-037faf9c0cc18a864 east-vpc	905418201986

igw-0abffcb8ef674e2a3 / east-igw

Details

Internet gateway ID	igw-0abffcb8ef674e2a3	State	Attached	VPC ID	vpc-037faf9c0cc18a864 east-vpc	Owner	905418201986
---------------------	-----------------------	-------	----------	--------	----------------------------------	-------	--------------

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 6 Hot weather Now 13:24 10-05-2025

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A specific Internet Gateway, 'igw-0abffcb8ef674e2a3 / east-igw', is highlighted. The 'Details' tab is active, displaying the Internet gateway ID (igw-0abffcb8ef674e2a3), state (Attached), VPC ID (vpc-037faf9c0cc18a864 | east-vpc), and owner (905418201986). The 'Tags' section shows a single tag named 'east-igw'. The left sidebar lists various VPC-related options like 'Your VPCs', 'Subnets', 'Route tables', and 'Internet gateways'. The bottom status bar indicates the region is United States (N. Virginia) and the user is srikanth kolluri.

Changed the Region to us-west-2:

The screenshot shows the AWS EC2 Instances page in the us-west-2 region. It displays one instance named 'WestInstance' with the ID 'i-0bc2959fdb99ce1d9', which is currently running. The left sidebar includes sections for 'Instances', 'Images', and 'Elastic Block Store'. A large dropdown menu on the right lists AWS regions, with 'us-west-2' selected. Other visible regions include N. Virginia, Ohio, N. California, Oregon, Mumbai, Osaka, Seoul, Singapore, Sydney, Tokyo, Central, Frankfurt, Ireland, London, Paris, and Stockholm. The bottom status bar shows the region is United States (Oregon) and the user is srikanth kolluri.

WestInstance created in us-west-2 region:

Instance summary for i-0bc2959fdb99ce1d9 (WestInstance) [Info](#)

Updated less than a minute ago

Instance ID	i-0bc2959fdb99ce1d9	Public IPv4 address	35.94.124.45 [Open address]
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-10-1-1-244.us-west-2.compute.internal	Private IP DNS name (IPv4 only)	ip-10-1-1-244.us-west-2.compute.internal
Answer private resource DNS name	-	Instance type	t2.micro
Auto-assigned IP address	35.94.124.45 [Public IP]	VPC ID	vpc-07f6ccabdbc59b84e (west-vpc)
IAM Role	-	Subnet ID	subnet-0797ef1807ab61fb (west-subnet)
IMDsv2	Optional ⚠️ EC2 recommends setting IMDsv2 to required Learn more	Instance ARN	arn:aws:ec2:us-west-2:905418201986:instance/i-0bc2959fdb99ce1d9
Auto Scaling Group name	-	Managed	false

us-west-2: Vpc,subnets and Internet gateway:

Your VPCs (1/2) [Info](#)

Last updated 1 minute ago

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-0231bad08326c56a7	Available	<input type="radio"/> Off	172.31.0.0/16	-
west-vpc	vpc-07f6ccabdbc59b84e	Available	<input type="radio"/> Off	10.1.0.0/16	-

vpc-07f6ccabdbc59b84e / west-vpc

Details [Resource map](#) [CIDRs](#) [Flow logs](#) [Tags](#) [Integrations](#)

VPC ID	vpc-07f6ccabdbc59b84e	State	Available	Block Public Access	<input type="radio"/> Off	DNS hostnames	Disabled
DNS resolution	Enabled	Tenancy	default	DHCP option set	dopt-07f3b43adc732440c	Main route table	rtb-0586b745f783a2b5f

VPC dashboard < Actions

vpc-07f6ccabdbc59b84e / west-vpc

Details		Info	
VPC ID	vpc-07f6ccabdbc59b84e	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-01e37f798620aaefc	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-0386b745f783a2b5f
		IPv6 pool	-
		Owner ID	905418201986

Resource map | CIDs | Flow logs | Tags | Integrations

Resource map Info

- VPC Show details Your AWS virtual network west-vpc
- Subnets (1)** Subnets within this VPC us-west-2a
- Route tables (2)** Route network traffic to resources rtb-0c49e8211c4fee569
- Network interface groups west-ii

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 13:28 10-05-2025

Subnet:

VPC dashboard < Actions

Subnets (1/5) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-026d30242d0b8bfac	Available	vpc-0231bad08326c56a7	Off	172.31.16.0/
-	subnet-0b484877d0317b083	Available	vpc-0231bad08326c56a7	Off	172.31.0.0/2
-	subnet-01303812b81e6e956	Available	vpc-0231bad08326c56a7	Off	172.31.48.0/
<input checked="" type="checkbox"/> west-subnet	subnet-0797ef1807ab61bfb	Available	vpc-07f6ccabdbc59b84e west-vpc	Off	10.1.1.0/24
-	subnet-05dc18f1493130a0f	Available	vpc-0231bad08326c56a7	Off	172.31.52.0/

subnet-0797ef1807ab61bfb / west-subnet

Details | Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

Details

Subnet ID	subnet-0797ef1807ab61bfb	Subnet ARN	arn:aws:ec2:us-west-2:905418201986:subnet/subnet-0797ef1807ab61bfb
IPv4 CIDR	10.1.1.0/24	State	Available
Availability Zone	us-west-2a	IPv6 CIDR	-
		Block Public Access	Off
		IPv6 CIDR association ID	-
		Network border group	vpc-07f6ccabdbc59b84e

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 13:28 10-05-2025

Internet Gateway:

The screenshot shows two views of the AWS VPC console. The top view displays the 'Internet gateways (1/2)' list, showing two entries: 'west-igw' (Attached, igw-0ec3d311268439f77) and another entry (Attached, igw-0f02c03da06d291af). The bottom view is a detailed view of the 'west-igw' Internet Gateway, showing its details (ID: igw-0ec3d311268439f77, State: Attached, VPC ID: vpc-07f6ccabdbc59b84e | west-vpc, Owner: 905418201986) and its tags (Name: west-igw).

Internet gateways (1/2) Info

Name	Internet gateway ID	State	VPC ID	Owner
west-igw	igw-0ec3d311268439f77	Attached	vpc-07f6ccabdbc59b84e west-vpc	905418201986
-	igw-0f02c03da06d291af	Attached	vpc-0231bad08326c56a7	905418201986

igw-0ec3d311268439f77 / west-igw

Details

Internet gateway ID igw-0ec3d311268439f77	State Attached	VPC ID vpc-07f6ccabdbc59b84e west-vpc	Owner 905418201986
--	-------------------	--	-----------------------

Tags

Key	Value
Name	west-igw

Now lets check whether the Nginx installed in both instances by accessing them:

Us-east-1 : ip : 54.161.127.28

Instance summary for i-022c5a42215de5747

Updated less than a minute ago

Instance ID: i-022c5a42215de5747

IPv6 address: -

Hostname type: IP name: ip-10-0-1-108.ec2.internal

Answer private resource DNS name: -

Auto-assigned IP address: 54.161.127.28 [Public IP]

IAM Role: -

IMDSv2: Optional
⚠️ EC2 recommends setting IMDSv2 to required | Learn more

Public IPv4 address copied

Private IP4 address: 10.0.1.108

Public IPv4 DNS: -

Elastic IP addresses: -

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendation

Auto Scaling Group name: -

Managed: false

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Hot weather Now

Feedback

https://54.161.127.28

12:54 ENG IN 10-05-2025

<http://54.161.127.28/>

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Not secure 54.161.127.28

Hot weather Now

Feedback

12:55 ENG IN 10-05-2025

Nginx Installed in us-east-1 region instance

Us-west-2 : ip : 35.94.124.45

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area is titled "Instance summary for i-0bc2959fdb99ce1d9". It displays various details about the instance, including its ID (i-0bc2959fdb99ce1d9), IPv6 address, Hostname type (IP name: ip-10-1-1-244.us-west-2.compute.internal), Answer private resource DNS name, Auto-assigned IP address (35.94.124.45 [Public IP]), IAM Role, IMDSv2 (Optional), Subnet ID (subnet-0797ef1807ab61fbf (west-subnet)), VPC ID (vpc-07f6ccabdbc59b84e (west-vpc)), Instance ARN (arn:aws:ec2:us-west-2:905418201986:instance/i-0bc2959fdb99ce1d9), Instance state (Running), Private IP DNS name (IPv4 only) (ip-10-1-1-244.us-west-2.compute.internal), Instance type (t2.micro), and other details like Private IPv4 addresses (10.1.1.244), Public IPv4 DNS, Elastic IP addresses, AWS Compute Optimizer finding, Auto Scaling Group name, and Managed status (false). A tooltip highlights the Public IPv4 address 35.94.124.45.

<http://35.94.124.45/>

The screenshot shows a web browser window with the URL "http://35.94.124.45" in the address bar. The page content is the standard Nginx welcome message: "Welcome to nginx!". Below it, there's a note: "If you see this page, the nginx web server is successfully installed and working. Further configuration is required." It also mentions "For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com." At the bottom, it says "Thank you for using nginx."



Nginx Installed in us-west-2 region instance